

# Distributed Case Based Reasoning for Intelligent Tutoring System: An Agent Based Student Modeling Paradigm

O. P. Rishi, Rekha Govil, and Madhavi Sinha

**Abstract**—Online learning with Intelligent Tutoring System (ITS) is becoming very popular where the system models the student's learning behavior and presents to the student the learning material (content, questions-answers, assignments) accordingly. In today's distributed computing environment, the tutoring system can take advantage of networking to utilize the model for a student for students from other similar groups. In the present paper we present a methodology where using Case Based Reasoning (CBR), ITS provides student modeling for online learning in a distributed environment with the help of agents.

The paper describes the approach, the architecture, and the agent characteristics for such system. This concept can be deployed to develop ITS where the tutor can author and the students can learn locally whereas the ITS can model the students' learning globally in a distributed environment. The advantage of such an approach is that both the learning material (domain knowledge) and student model can be globally distributed thus enhancing the efficiency of ITS with reducing the bandwidth requirement and complexity of the system.

**Keywords**—CBR, ITS, Student Modeling, Distributed System, Intelligent Agent.

## I. INTRODUCTION

THE major challenge in teaching is to improve both instructional productivity and learning quality for large and diverse population of students under real world constraints such as limited financial resources and insufficient number of qualified instructors [1]. Different researches in the field suggest that students who are engaged in learning through intelligent tutoring processes are more likely to achieve success [2].

Since mid 90's, few educational system models were web based and used ITS with student modeling in distributed style [2][8]. In past decade, researchers from different disciplines have come out with systems which define and classify different teaching and learning styles in distributed

environment [2][9].

In this paper we conceptualize a Case Based Distributed Student Modeling (agent based) ITS architecture to support student-centered, self-paced, and highly interactive learning. In this system the first step in building an effective learning environment is building a Case Base where the system maintains a rich set of cases (scenario) of student's learning pattern, and employs an efficient and flexible case retrieval system. This maximizes the interactivity between the ITS and the students; and customizes the learning process to the needs of an individual student [3]. The system must use the student's learning profile such as learning style and background knowledge in selecting, organizing, and presenting the learning material to support case based learning. It also supports personalized and more intensive interaction between the student and the ITS [4]. Distributed CBR based student modeling enables adaptive delivery of educational contents and facilitates automatic evaluation of learning outcomes[1][9]. The system also incorporates a new approach to course content organization and delivery, which can be developed, based on distributed and agent based instructional components. Instructional components represent the customized interactive presentation of any topic of a subject or different subjects.

## II. PROPOSED SYSTEM FOR DISTRIBUTED CASE BASED REASONING

The proposed system is based on finding a case that is similar to the learning domain of a past student in a distributed environment.

The system consists of a number of specialized agents with different expertise. Each student has a unique *personal agent* (student profiler) that manages the student's personal profile, including knowledge background, learning style, interests, courses enrolled in, etc. The other two agents in the system are *Teaching Agent* and *Course Agent* and they communicate with each other through different communication channels situated in a distributed environment.

The model of communication between agents shown in Fig. 1. A web based course is supported combinedly by a *teaching agent* and *course agent* that manage course material and course-specific teaching techniques and student modeling strategy. Multiple course agents exist on distributed sites to

O. P. Rishi is with the Birla Institute of Technology, Extension Centre Jaipur, BISR campus, 27 Malviya Industrial Area, Jaipur 302017, India (phone: + 91 94142 58030; fax: + 91 141 2751601; e-mail: omprakashrishi@yahoo.com / oprishi@bitmesra.ac.in).

Rekha Govil is with the Apaji Institute of Mathematics & Applied Computer Technology, Banasthali Vidyapith, Rajasthan-304022, India (e-mail: grekha@banasthali.ac.in.).

Madhavi Sinha is with the Apaji Institute of Mathematics & Applied Computer Technology, Banasthali Vidyapith, Rajasthan-304022, India (e-mail: smadhavi@jp.banasthali.ac.in).

provide greater efficiency, flexibility, and availability.

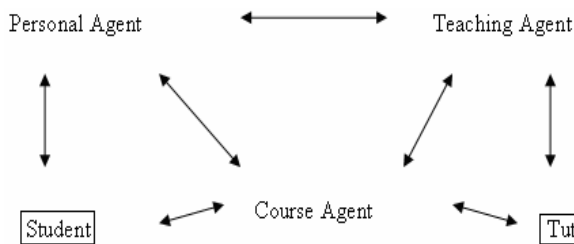


Fig. 1 Communication model among Agents

The *teaching agents* can talk to any *course agent*, and often choose one nearby for better performance. The *course agents* also act as mediators for communication among students and tutors. A *teaching agent* interacts with a student and serves as an intelligent tutor of a topic or course. From a *course agent*, each *teaching agent* obtains course material and course-specific teaching techniques and then tries to teach the material in the most appropriate form and pace based on the background and learning style of the student. Lecture notes, presentations, multiple examples, with different difficulty level are used to make difficult concepts and operations easy to understand.

The following characteristics specify the attractiveness of this CBR based distributed student modeling:

- 1) In the learning environment students, tutors and corresponding student learning material are distributed across the entire network. Similarly, the potential users as student or as tutor are also widely distributed.
- 2) The student's behavior, corresponding background knowledge, and skills are dynamic, and accordingly the learning material and teaching methodology of the ITS are also required to be dynamic in nature i.e. they depend on case to case basis [5]
- 3) Students have different backgrounds, learning attitude and personalities. Students generally attempt to register in various courses at the same time. In this case coordinating learning on different topics for each student enriches the learning experience with in different environment. This is possible using CBR based distributed student modeling through personal agent.

### III. STUDENT MODELING IN DISTRIBUTED ENVIRONMENT

The process of student modeling [7] is shown in Fig. 2. The following activities take place during the student modeling when the student interacts with the system.

- 1) Selection of topic by the student and getting student's background by presenting problems to the student.
- 2) Analyzing the student's response by the system
- 3) Selection of case by the system based on the response
- 4) Adaptation of the case by the system (if new then leave the case)
- 5) Achieving the knowledge component of the student model through case retrieval

- 6) Generation of teaching strategy by the system
- 7) Presenting the next problem / content to the student

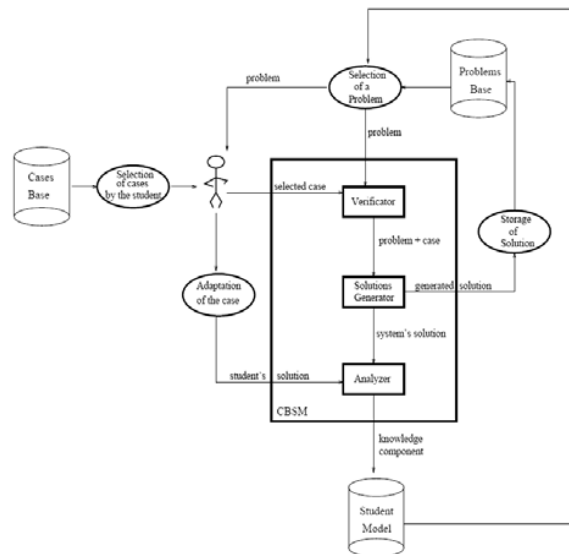


Fig. 2 Process of Student Modeling [7]

Any student can register as a user at any node over the entire network. The student can select any topic for learning. This can be done through navigation of topics provided by the ITS. However, before allowing for navigation, the *student profiler agent (personal agent)* of the system checks the past record of the performance of the student to present the right content, he ITS.

If the performance of the student is not satisfactory then the appropriate remedial content is presented to the student. Only after the learning objectives are achieved, the student is allowed to move to a new topic. If any topic (case) is not available at the local node then the system searches it from other nodes within the network. This is done with the help of *teaching agent* and a topic index that is maintained at every node. The agent is responsible for updations in the *index*. If the case is not found on the entire network then the *teaching agent* interacts with the human tutors for "plug-in topics". These new plug-in topics are stored at least on two nodes and on one more nearest node for fault tolerance. This is the responsibility of the *course agent* to decide as to where should the duplication be made. The *course agent* is responsible for course management.

The design of such a case based student modeling system requires algorithms for the following operations:

- 1) Case creation in active case base.
- 2) Case indexing in active case base
- 3) Update case in case base
- 4) Adaptation of cases in the case base
- 5) Case storage
- 6) Pattern matching of case in case base or Retrieval of cases from case base
- 7) Checking duplicate case from case base
- 8) Case Deletion from case base and storing the deleted case

in to archive for future reference

Not only the student modeling but the tutor module of the ITS is also distributed.

The *teaching agent* does not interact directly with a student but through *personal agent* and serves as an intelligent tutor. The basic components of a teaching agent are a domain expert module and pedagogy. The architecture of *teaching agent* is shown in Fig. 3.

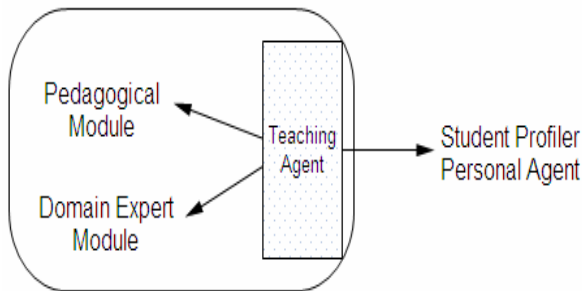


Fig. 3 Teaching Agent

The domain expert module is responsible for storing and managing content, exercises, assignments, and questions with the help of *course agent*. It contains a problem generator, a problem solver, an explanation generator, and a domain knowledge base. The pedagogy module determines the timing, style, and content of the *teaching agent's* interventions. It is a rule-based production system that uses the student model and pedagogical knowledge to determine the appropriate actions. The student profiler or *personal Agent* provides a model of a student based on learning style, knowledge background, and interests. It may also incorporate the information about student's learning profile gathered through dialogue with the student such as the actions the student performed and the explanations which student made [1].

#### IV. ARCHITECTURAL DESIGN OF CBR BASED STUDENT MODELING IN DISTRIBUTED ENVIRONMENT

CBR Based student modeling system can be divided into three parts. A Case Based ITS (Intelligent Tutoring System), a student interface module and the instruction processing engine or agent communication system. The ITS is divided into a student module, a tutor module with expert module (human-tutor interaction), and an interface. The student module consists of a Case Base, which is based on an overlay or bug part library system [2].

The interfaces of student module, tutor module and case based student modeling reside on every node of the network. *Teaching Agent* indexes all the cases globally. *Personal agent* or student profiler maintains static background profile (such as GPA or percentage, majors, interests and course taken) and the student's dynamic profile based on their interaction [6]. The agent is responsible for indexing all the registered or aspirants students (learners) with their personal profiles globally. The *course agent* manages the course material as per student's requirement and *teaching agent* decides the topic or

lessons, to be supplied to the students according to student's performance. The student modeling keeps the record in case base for future reference and manages the student performances and course material accordingly. The nodal architecture of the proposed system is shown in the Fig. 4.

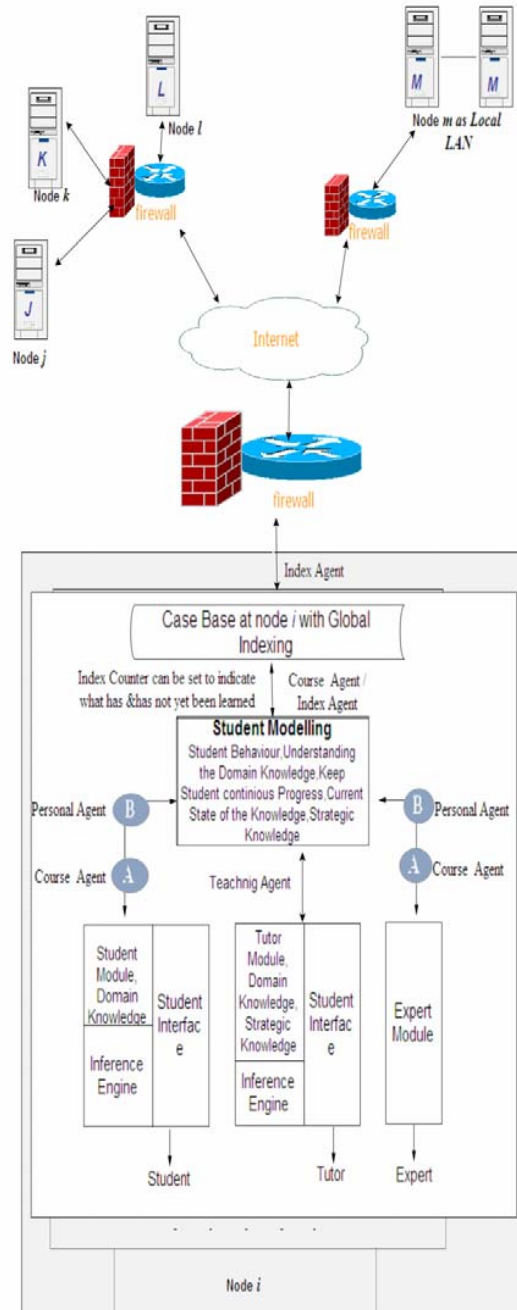


Fig: 4 Design Architecture for CBR Based ITS in Distributed & Agent Environment

Let us think of a scene of learner at node *i* interacting with the ITS. Node *i* can be a stand alone user or can be a school with LAN. If the student is the first time user then he/she is

registered with the system. The *personal profile agent* maintains the user profile. The session starts and student modeling activities initialize.

The past performance is evaluated first and thereafter *teaching agent* suggests the topic. The component of *teaching agent* is shown in Fig. 5.

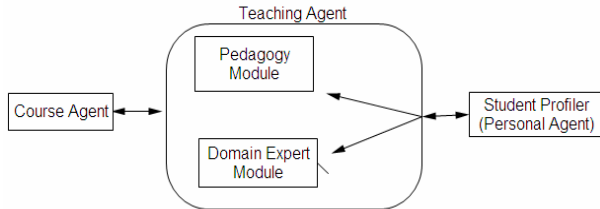


Fig. 5 Teaching Agent

The *Course agent* retrieves the appropriate topic first at the local node *i* and presents the material to the student. If the appropriate topic content is not found at node *i* the *teaching agent* then searches the appropriate content from the entire network with the help of the course index's. The students' learning is accessed by presenting questions to the student. The response is analyzed and the *personal agent* matches the leading style with the cases present in the case base which is stored in distributed manner with the help of global case index. If the case is new i.e. there is no similar case stored within entire network then this case is stored as new at node *i* and its one more copy is stored at some another node for fault tolerance. This is the responsibility of the *course agent* to manage this storage. Thus the whole system is managed in the distributed environment with just three agents:

- 1) Student profiler or Personal Agent,
- 2) Teaching Agent and ,
- 3) Course Agent.

The major characteristics of the System are:

- 1) The system is fully distributed (not bounded with any network topology) i.e. domain knowledge and strategic knowledge both are distributed.
- 2) Reduces the need of large storage spaces at the user's site, to store all the cases.
- 3) Redundancy (duplicacy) is maintained for fault tolerance. Load balancing of the cases is achieved at each node by the storage management. Later one can research on optimum redundancy parameter.
- 4) Case indexing is fully redundant on all nodes.
- 5) If the node is a LAN, the domain knowledge can be localized and case base can be global

## V. FUTURE WORK & CONCLUSION

A prototype of the Distributed Case Base Reasoning for Intelligent Tutoring System is currently under development where a 'C' Language Tutor is being designed and developed. The implementation result shall be presented in a sequential publication.

Future work under this research, will focus on the following issues:

- 1) Design of a conceptual Student Modeling module, to

provide decisions for the best-case retrieval from the Case Base.

- 2) Implementation of CBR for student modeling and ITS in distributed Environment.
- 3) Design and development of agents
- 4) Algorithm for maintaining redundancy (duplicacy) for fault tolerance incorporating load balancing.

## ACKNOWLEDGMENT

The authors are grateful to the anonymous referees for their constructive and helpful comments.

## REFERENCES

- [1] Yi Shang, Hongchi Shi, and Su-Shing Chen Deptt of computer engineering and science, University of Missouri-Columbia; An Intelligent Distributed Environment for active Learning, may 1-5, 2001 hong Kong. ACM 1-5113-348-0/01/0005.
- [2] Ang Yang, Ray Kemp, Kinshuk; 2001, Web based intelligent tutoring system.
- [3] Sinha M. & Govil R.; 1995, PRABODH: A Distributed online Hindi Grammar Teaching Learning System. IJMS vol 11, No.3.
- [4] Brusilovsky, P., Schwarz, E. Weber, G. ELM-ART: An intelligent tutoring system on World Wide Web. Intelligent Tutoring Systems Lecture Notes in Computer Science, Vol. 1086. Berlin: Springer Verlag. (1996) 261-269.
- [5] Aamodt, A.; 2005, Knowledge Intensive Case-based reasoning and Intelligent tutoring system.
- [6] Leen-kiat Soh; 2005, Incorporating Intelligent tutoring system in to CS1, SIGCSE06.
- [7] Shiri M. E. , Aimeur A. E. and Frasson C.; 1998, Student modelling by Case Based Reasoning, Lecture Notes in Computer Science, vol 1452.
- [8] C. Burger And K. Rothermel, A Framework to Support Teaching in Distributed Systems, ACM Journal of Educational Resources in Computing, Vol. 1, No. 1, Spring 2001.
- [9] Amruth N. Kumar, Generation of Problems, Answers, Grade, and Feedback – Case Study of a Fully Automated Tutor, ACM Journal of Educational Resources in Computing, Vol. 5, No. 3, September 2005.



**O. P. Rishi** received the M.Tech. Degree in computer Science from Birla Institute of Technology, Mesra Ranchi (India) in 2000, the M.Sc. Degree in Mathematics from University of Rajasthan in 1995, He is a Ph. D. Student in Computer Science at Banasthali Vidyapith, Banasthali Rajasthan. His current research involves Case Base Reasoning for Intelligent Tutoring Systems and intelligent systems.



**Dr. Rekha Govil** is professor, computer science and Dean Apaji Institute of Mathematics & Applied Computer Technology, Banasthali Vidyapith. She has more than twenty years experience of teaching and research, has executed a number of research projects. She is into guiding Ph.D. research and till now ten students have been awarded Ph.D. Her research interests are very wide and include AI, smart instrumentation and educational computing.



**Dr. Madhavi Sinha** received the Ph.D. Degree in Computer Science as the topic on the intelligent tutoring system from Banasthali Vidyapith, after M.Sc. degree in Computer Science from the same university. Her research interest is in NLP and other artificial intelligence application.